Maturity of Software Modelling and Model Driven Engineering: a Survey in the Italian Industry

Maturity of Software Modelling and Model Driven Engineering: a Survey in the Italian Industry

Federico Tomassetti, Marco Torchiano
Department of Control and Computer Engineering
Politecnico di Torino, Italy
Email: {federico.tomassetti,marco.torchiano}@polito.it

Alessandro Tiso, Filippo Ricca, Gianna Reggio
DISI
Università di Genova, Italy
Email: {alessandro.tiso,filippo.ricca,gianna.reggio}@disi.unige.it

***Abstract*—Background: The main claimed advantage of Model-driven engineering is improvement in productivity. However, few information is available about its actual usage during software development and maintenance in the industry.**

Objective: The main aim of this work is investigating the level of maturity in the usage of software models and of Model-driven engineering in the Italian industry. The perspective is that of software engineering researchers.

Method: First, we conducted an exploratory personal opinion survey with 155 Italian software professionals. The data were collected with the help of a web-based on-line questionnaire. Then, we conducted focused interviews with three software professionals to interpret doubtful results.

Results: Software modelling is a very relevant phenomenon in the Italian industry. Model-Driven techniques are used in the industry, even if (i) only for a limited extent, (ii) despite a quite generalized dissatisfaction about available tools and (iii) despite a generally low experience of the IT personnel in such techniques.

Limitations: Generalization of results is limited due to the sample size. Moreover, possible self-exclusion from participants not interested in modelling could have biased the results.

Conclusion: Results reinforce existing evidence regarding the usage of software modelling and (partially of) Model-driven engineering in the industry but highlight several aspects of immaturity of the Italian industry.

I. Introduction

Model-driven engineering (MDE), model-driven architecture (MDA), and model-driven-development (MDD) are different "declensions" with slightly different meanings of a software development methodology defined in the last decade and based on the concept of model [11]. They raise the abstraction level from the code to the models, which are considered primary assets. In practice, models are transformed and the code is generated from them by means of (semi) automatic transformations. Alternatively, the models can be directly executed/interpreted. In this latter case, they are named executable models (e.g., represented by means of Executable UML [13]). In the following, we will address all these related techniques with the abbreviation MD* [19].

In 2003, Kleppe, Warmer and Bast wrote: "We feel that MDA could very well be the next major step forward in the way software is being developed. MDA brings the focus of software development to a higher level of abstraction, thereby *raising the level of maturity of the IT industry*" [11]. Still, they asserted: "*the primary advantages of MDA are improvements in productivity, portability, maintainability and interoperability*" [11].

Several years after those claims, we are interested to understand if and how the Italian industry has adopted (and adopt) models and Model driven techniques during software development and maintenance tasks. For this reason, we decided to evaluate in a systematic way MD* diffusion, maturity and effectiveness in an industrial context by means of an exploratory personal opinion survey [1] with IT professionals.

The evidence-based answers we provided to the research questions on MD*, by means of this survey, hold a value in itself as important knowledge assets in the software engineering field. In addition, they bring important implications in the practice of both software development and education/training. Moreover, conducting a survey on this topic can help to highlight possible immaturity aspects of software modelling and MD* usage in the Italian industry. We believe that promoting improvements and strategies without truly understanding the MD* immaturity aspects is, in essence, putting the cart before the horse.

Here, we only present some selected results of this study, concerning mainly MD* maturity. Some other aspects, not considered here, e.g., languages used, have already been sketched in [18].

The survey is the most common method of gathering information. It can be: a face-to-face interview, telephone, mail/e-mail or Internet survey. We selected the latter option, putting on-line a questionnaire and inviting people to answer, because an Internet survey is generally the most cost effective interview method [20], even if it presents well-known limitations/problems (e.g., sampling bias and difficulties in designing clear, unbiased and unambiguous questionnaire questions) [8].

After data analysis, to corroborate the obtained results and better understanding some doubtful results we got, we asked the help of three expert software professionals and we conducted with them a debriefing session[1]. These (one-to-one) interviews helped us to better interpret the obtained results.

The paper is structured as follows. Section II presents

[1]Debriefing sessions are semi-structured conversations with an individual that has just experienced an event. Usually, a debriefing session is used in software engineering after an empirical study to receive information and understanding better some results.

the relevant aspects of the conducted survey such as the questionnaire design, sample identification and survey preparation/execution. Section III presents the findings about MD* maturity. Section IV discusses the obtained results, and section V summarizes the point of view of the three experts about some difficult-to-interpret results. Section VI presents the unavoidable threats to validity and section VII sketches related works. Finally, section VIII concludes the paper.

II. The Survey

We conducted an industrial survey to investigate to what extent and how Italian companies deal with software models and MD*. To implement this survey, we (i) used the same framework of [17], [**?**] (based on [9]), (ii) followed as much as possible the suggestions given in [1] and (iii) adopted an on-line questionnaire to collect information.

In this report, for space reasons and to better describe the obtained findings, we will consider only the main goal of the survey: *understanding and documenting if and how MD* is currently used in industry* (for the interested reader, the secondary goals, such as for example languages and notations used in software modelling and MD*, are considered more specifically in [18]). Thus, here we will consider only one aspect of our survey: the maturity in the usage of models and MD* in software development and maintenance.

A. Questionnaire Design

To design the questionnaire we adopted the Goal-Question-Metric approach [2] and followed a standard schema [3]. From the goals we derived the questions and the metrics necessary to answer them. In other words, the questionnaire has been developed to directly address the goals of the study.

In order to harvest as many responses as possible we limited the time to complete the questionnaire to approximately 10-15 minutes (long questionnaires get less responses than short ones [20]) and designed it accordingly[2]. The questionnaire contains 31 questions, both open and multiple-choice, though the effective number of questions actually presented to a respondent depends on his (her) level of usage of MD* and modelling (e.g., respondents non using modelling in their software process are required to answer eight questions only).

The questionnaire consists of four sections. Section 1, common to all respondents, characterizes their organization. In particular, it collects: business domain, organization size, respondent's group/business unit size and experience of unit members.

Section 2 asks about the processes adopted: the kind of projects conducted and their average duration, if the respondent uses models and with which goal. In this section the most important question is DEV08[3] that reads as follows: "*Are models used for software development in your organization?*" By model we mean both diagrams and text artifacts created using either general purpose modelling languages (e.g., UML) or Domain Specific Languages (DSLs).

Sections 3 and 4 — administered only to subjects who answered "yes" to question DEV08 — collect information concerning modelling and MD* usage. There are 21 questions about modelling languages, notations, tools and processes. For instance, we asked about years use of MD*, experience in MD* and percentage of projects in which models are used with respect to all the projects undertaken. Moreover, we also collected information about code generation (e.g., degree/percentage of code generated with respect to the final product), execution of models by means of specialized interpreters and usage of automatic transformations (model-to-model and model-to-text).

The complete adopted questionnaire (in Italian) is available at: *http://softeng.polito.it/tomassetti/MDQuestionnaire.pdf.*

The questionnaire was introduced to the contacts with a brief table summarizing goal and motivation of this study and it was accompanied with a cover letter briefly introducing our research project. In the cover letter we tried to summarize: what the purpose of the study is, why it should be of relevance to the participants and why each individual's participation is important [1].

We decided to avoid any form of material incentives for participation. However, to motivate the professionals, we promised to provide to all participants a report containing the analyses and the obtained results. In addition, we offered to invite all the participants to a workshop about MD* where the obtained results will be presented in aggregate form.

B. Population and Identification of the Sample

The first step to conduct a survey is defining a target population. The target population of our study consists of software development teams or business units. In order to get information about the target population we defined a framing populations consisting of Italian software professionals (e.g., project managers, architects, developers) whom we asked to answer our questions.

To sample the population we applied both probabilistic (random sampling) and non-probabilistic (convenience sampling) methods [1]. More in detail, the sampling was performed in five ways: (i) using the Commerce Chamber database and randomly selecting some contacts; (ii) as a convenience sampling relying on the network contacts of the two research units involved (Torino and Genova); (iii) sending invitation messages on mailing lists concerning programming and software engineering; (iv) publishing a note on an on-line magazine for developers (programmazione.it); and (v) advertising it on a large Italian conference for developers (CodeMotion 2011).

In total, we obtained 155 complete responses to our survey, thus the context of our survey consists of a sample of 155 Italian software professionals[4]. Unfortunately, we did not know exactly how many people have been reached by our invitation

[2]It turned out the actual time for completing the questionnaire was on average less than six minutes.

[3]A unique identifier composed by question type and number was assigned to each question.

[4]It is difficult to precisely identify the percentage of participants coming from selected invitation. Approximately, they should be 40-45 out of 155.

Dimension	ID	Question	Type of Data	Findings
Automation	MOD14	Which percentage of code is generated starting from models?	Interval	42% of generated code
	$MOD14^*$	Derived from previous: $MOD14^* \leftarrow MOD14 > 0$	Yes/No	44% adopters of modelling
	MOD15	Which parts/tiers of the system are automatically generated? (e.g., presentation, business and data logic)	Nominal	Mostly SQL scripts, presentation logic and architectural code
	MOD16	Are models executed (interpreted) at run-time?	Yes/No	16% adopters of modelling
	MOD18	Are transformation languages (e.g., ATL) used? (model-to-model and model-to-text transformations)	Yes/No	10% adopters of modelling
Tools	MOD19a	Are specialized editors or other supporting tools developed for managing/creating models? (e.g., using Xtext that is a tool for developing textual DSLs)	Yes/No	16% adopters
	MOD19b	Which technologies are used to develop specialized editors or other supporting tools?	Nominal	GMF (6 users) and Xtext (4 users)
	MOD20	Are versioning systems used to manage models?	Yes/No	53% adopters
	DEV09	What are the problems hindering or preventing modelling and MD* (if any)?	Nominal	35% tools unavailability/immaturity/cost
Experience	MOD21	Since how many years is modelling used?	Ordinal	Median 5 years, average 5.35 years
	MOD22	In how many projects has modelling been used in the last 2 years?	Interval	Median 30%, average 44%
	MOD24	Since how many years is MD* used?	Ordinal	Median 4 years, average 4.30 years
	MOD23	In how many projects has MD* been used in the last 2 years?	Interval	Median 30%, average 39%

TABLE I
SUMMARY DATA ABOUT MATURITY

messages and advertisements, so we can not calculate response rate (the same problem is present in [12]).

It is important to emphasize that in our sample the correspondence between respondents and companies is not one-to-one. In particular, we could have more questionnaires compiled by professionals belonging to the same company but employed in different business units/groups. Clearly, this is more probable in large companies hosting several business units and work groups.

C. Data collection

We collected data through an on-line questionnaire created by means of the LimeSurvey survey tool[5]. Web-based questionnaires, compared to paper-based questionnaires or email-based questionnaires, allow an easier data entry from the respondent perspective and a simpler data collection from the researcher perspective. In general, it has been observed that Web-based questionnaires guarantee high return rates [9].

D. Survey Preparation and Execution

The survey was put on-line from the 1st of February 2011 until the 15th of April 2011 (two and a half months). The procedure followed to prepare and administer the questionnaire, and collect the data is made up of the following five main steps.

Preparation and design of the questionnaire. We first prepared the questionnaire. Then, we conducted three different pilots with software professionals before putting on-line the final questionnaire (survey instrument evaluation phase). Usually, pilot studies are intended to identify any problems with the questionnaire itself improving the validity of the instrument [1]. According to the feedback obtained, minor changes at the questionnaire were made.

On-line deployment. Once ready, the questionnaire was uploaded to the LimeSurvey survey tool to permit the automatic collection of data.

Invitation to participate. Organizations were sampled as detailed above. Once the contact persons were identified, we invited people, via email, to register to the survey and to complete the on-line questionnaire. We also broadcast invitation on selected mailing lists and on-line magazines/conferences including in the message a link ("click here to take the survey") to a registration form where the participants could register themselves and compile the questionnaire.

Monitoring. During the data capture phase, we monitored the progress of the questionnaire submission. This allowed us to send selective reminders to contacts who did not respond or did not completed the questionnaire. Some people that reported some difficulties about the questions, because of internal policies of the company or because involved in very different projects with different companies at the same time, asked to us some clarifications about the questions.

Data analysis. After questionnaires have been collected, analyses were performed. Given the nature of this survey, that is mainly descriptive (it describes some condition or factor found in a population in terms of its frequency and impact) and exploratory, we applied more descriptive statistics and showed our findings mainly by means of graphs/charts.

As a tool to analyse the maturity, we used polar charts. A polar chart is a chart constituted of a sequence of equi-angular spokes representing one of the variables of interest[6]. The length of a spoke is proportional to the magnitude of the variable for the data point relative to the maximum magnitude of the variable across all data points. A line is drawn connecting the data values for each spoke.

In some cases, we applied statistical tests to looks at the relationship existing among factors. Statistical differences between groups in the data set were verified by means of non-parametric tests (mainly due to the scale of the variables). In particular, we used Mann-Whitney tests for testing the difference between two groups and the Kruskal-Wallis test for three or more groups [**?**] (Chapter 37).

III. FINDINGS

We analyse the subset of questions in the questionnaire that are relevant to address the main research question (RQ): "*What is the level of maturity of MD* in the Italian Industry?*". Such

[5]http://www.limesurvey.org/

[6]See http://en.wikipedia.org/wiki/Polar_chart.

	Category	No. Respondents
Experience	[0,2] years	29
	(2,5) years	21
	[5,10) years	39
	[10+ years	16
Company size	Micro (1-10 employees)	28
	Small (11-50 employees)	14
	Medium (51-250 employees)	20
	Large (250+ employees)	43

TABLE II
EXPERIENCE LEVELS AND COMPANY DIMENSION

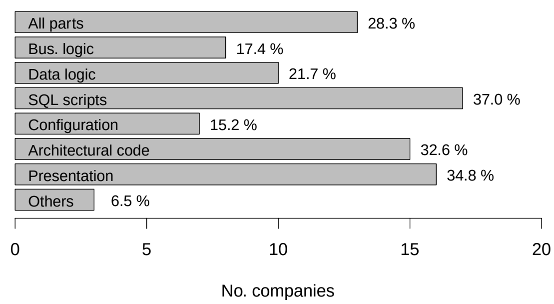

Fig. 1. Parts/tiers of the application generated from models

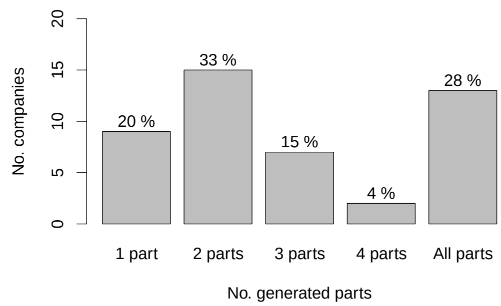

Fig. 2. Number of different generated parts/tiers of the system per company

questions are reported in Table I with the identifier, the type of data of the answers, and a summary of findings.

Taking inspiration from [14], we considered three dimensions to measure maturity (*Level of automation*, *Tools* and *Acquired experience in modelling and MD**) and grouped the selected questions accordingly (see Table I). We assumed that higher levels of automation in the development process (e.g., measured as percentage of generated code) correspond to higher MD* maturity of the company. In the same way, development of modelling/MD* tools (such as for example graphical editors to draw models or tools for developing textual DSLs) and significant employees' experience in modelling and MD* (e.g. measured in years) equate to high maturity of the company.

A. Diffusion of modelling

In total, we obtained 181 responses: 155 complete questionnaires and 26 incomplete ones (discarded before the analysis phase). Among them, we found 105 adopters of modelling (68% of the valid sample), i.e., respondents that at least, sometimes, produce models or apply MD* techniques (DEV08).

Table II breaks the modellers (i.e., only the adopters of modelling) according to company size and experience in modelling. From that Table, we can see that the modellers are uniformly distributed, with a slight preponderance of experienced professionals working for large companies.

In the following analysis of maturity, we will consider all respondents that perform modelling, considering the sheer production of models the level zero of maturity.

B. Level of automation

In order to evaluate the level of automation we considered three indicators — code generation, model execution and usage of model transformations — corresponding to four questions of the questionnaire (see Table I). We found that 50 participants (48% of the adopters of modelling and 32% of the entire valid sample) actually employ one or more of the three key techniques.

Code generation: we found that code generation is often used (MOD14*), indeed 46 developers, 44% of the adopters of modelling (nearly 30% of the entire valid sample), generate portions of code in an automatic way.

The amount of application code that is generated is 42% as mean and the median is 30%. Moreover, there is a large variability: the lower and the upper quartiles capture a wide range between 10% and 80% of code generated.

Automatic generation targets different parts/tiers of the system (MOD15), the most common ones are SQL scripts, presentation logic and architectural code (see Figure 1 for the complete statistics). In addition, we counted how many distinct parts/tiers of the system are automatically generated in each respondent's company. Results are shown in Figure 2. The most common category is "2 parts" (33%). It is also interesting to note that 28% of respondents, among the adopters of modelling, are used to target all the parts/tiers in the application.

Model execution: 17 companies out of 105 adopt model execution, i.e., 16% among the adopters of modelling (MOD16). It is interesting to note that companies executing models generate automatically a significantly larger amount of code than companies not executing them. The relation is shown in Figure 3 (left) by means of boxplots. The difference is statistically significance as Mann-Whitney test p-value is < 0.001. Adopters of model execution tend to generate approximately 50% of the code (median), while non-adopters have on average a code-generation close to zero. Apparently, the respondents in our sample do not consider model execution and code generation as mutually exclusive alternatives. Instead, they tend to use both techniques. This result deserves further investigation.

Model transformations: 11 companies out of 105 perform model transformations, i.e., 10% among the adopters of modelling (MOD18). Adopters of model transformations tend to generate 80% of the code (median) while non-adopters rarely generate significant amounts of code. The difference is statistically significant (Mann-Whitney p-value < 0.001) and is illustrated in Figure 3 (right).

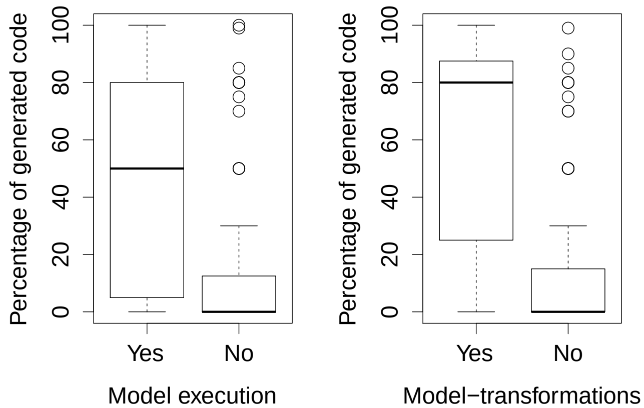

Fig. 3. Relation between model execution and code generation (left) and relation between usage of model transformations and code generation (right)

C. Tools

Four questions of the questionnaire concern tools (see Table I). They address three aspects: development of supporting tools (e.g., specialized editors), versioning, and adoption problems.

Supporting tools: 17 companies out of 105 have developed custom editors or other tools for managing and/or creating models. They are 16% of the adopters of modelling (MOD19a). The most used technologies (MOD19b) for developing these tools are GMF (6 users) and Xtext (4 users). Both those technologies are included in the Eclipse Modelling Project: Xtext assists the development of textual DSLs while GMF supports the creation of graphical DSLs. While developing specialized editors could be an indicator that the company is willing to invest in MD* and considers it relevant, the lack thereof does not necessary imply that the MD* approach adopted is immature: the organization could adopt a "ready-made MD* solution" developed externally (e.g., WebRatio [**?**]).

Versioning: 56 companies out of 105 use versioning systems to manage their models, that is just 53% among all the modellers (MOD20). This percentage can be considered an element of immaturity for the community of modellers. The reasons of the low usage of versioning tools should be better understood with further studies (e.g., by means of specific interviews focused on that aspect). Maybe, in some cases the models are not concurrently developed or they do not need to be maintained because they are just used to generate the skeleton of the application or to support discussion. Organizations which do not generate code, use versioning in the 46% of the cases while companies generating more than 80% of code use it in 75% of the cases.

Adoption problems: Among respondents who reported problems hindering or preventing modelling adoption (DEV09), 35% of them reported one or more problems related to supporting tools (for the other problems see [18]). Reasons of dissatisfaction about the tools are: their unavailability (23 respondents, 17% of the ones reporting problems) or immaturity (14, 10%)[7], their cost (14, 10%) or fear of vendor lock-in (13, 9%).

Out of 31 respondents which complained about the available tools, four of them developed some tools in-house.

D. Experience

Four questions in the questionnaire are devoted to the experience dimension (see Table I). They capture information regarding modelling and MD* employees experience measured in years (staff experience) and percentage of projects where modelling and MD* are adopted respectively (organization experience). The responses are summarized in Figure 4.

Staff experience: 40% of users using modelling in our sample have an experience in the range (2,5] years and 30% in the range (5,10] (MOD21). The experience in MD* is lower: 65% among modellers have no experience at all and 20% have 2 years or less of experience (MOD24). Figure 4 (middle) shows the complete distributions for experience in modelling and MD*. The experience in modelling appears to be distributed according to a normal distribution centred around the interval of (2,5] years of experience. Instead, the experience in MD* has a distribution strongly skewed towards the zero.

Organization experience: The teams of our sample use modelling in 44% of the projects on average (MOD22)[8]. Adopters of modelling use MD* in 39% of the projects on average (MOD23). In both cases the time-frame considered is the last two years. Both modelling and MD* are used in more projects as the experience of the respondents in the field grows (see Figure 4, up for modelling and down for MD*). For example, see Figure 4 (up), modellers with an experience in the range (0,2] years adopt modelling only in the 20% of the projects (median) while modellers with more than 10 years of experience adopt it in the 80% of the projects (median).

The correlations between years of experience and proportion of projects adopting modelling and MD* respectively are statistically significant. In both cases the Kruskal-Wallis test returned a p-value < 0.001.

E. MD maturity by company size and experience in modelling*

Among the 13 indicators we analysed above (see Table I), we identified a subset of them to be investigated further, having in mind the goal of observing how different types of companies score in terms of maturity. As categorization criteria, we used: (i) company size and (ii) years of experience in modelling.

We decided to consider only boolean indicators (Yes/No answers) since they enable an immediate quantification of the maturity level of a group of companies: the percentage of companies having a positive indicator provides a measure of maturity in that group. Conversely, we discarded, for this analysis, the indicators measured with interval, nominal and ordinal metrics. For the interval metrics it is difficult to define

[7]The former is the number of respondents, the latter is the percentage

[8]The values were calculated considering only users with more than zero years of experience respectively in modeling and MD*

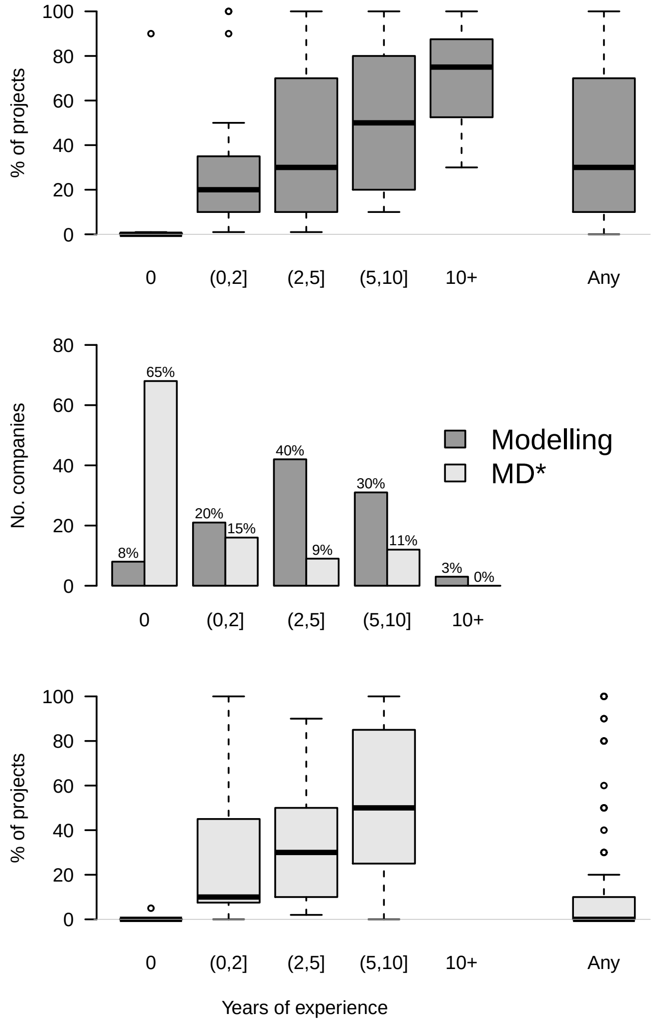

Fig. 4. Staff (middle) and Organization experience (up and down) in modelling and MD*

one or more thresholds to assess the maturity level, e.g., is 42% of average code generated low or high? The same reasons hold, all the more so, for ordinal and nominal metrics.

Finally, the selected indicators are: MOD14* (CodeGeneration), MOD16 (ModelExecution), MOD18 (Transformations), MOD19a (Specialized editors), and MOD20 (Versioning).

As a tool to analyse the maturity from those five perspective at once, we used polar charts. We recall that each line in a polar-chart represents a quantitative synthesis of maturity. It was calculated as the mean of the values for that indicator between the respondents of a given group.

Figure 5 (left) shows the maturity along the five dimensions for companies grouped into different ranges of size, each size category corresponding to a different line style. By looking at the enclosed areas, micro companies (10 employees or less) appear to use a more mature approach in MD* than larger companies (indeed the area is bigger). The percentage of companies adopting code generation varies significantly as the size of company: micro companies and large companies, the latter to a lesser extent, adopt code generation more often than small and medium companies. The same applies to model transformations. In terms of model execution and specialized editors, micro companies perform better than the larger-sized companies. Finally, in terms of model versioning large companies adopt it slightly more than micro companies, while small and medium sized companies embrace it half as frequently.

Figure 5 (right) shows the five indicators based on the experience in modelling. The picture here, is less clear cut. Companies in the (5-10] range adopt slightly more often model transformations and execution and more often code generation than companies in the 10+ range; while the latter developed more often specialized editors. Companies with shorter experience adopt such techniques less often. Finally, we observe a natural evolution, as experience grows, from a reduced adoption of versioning up towards more diffuse adoption.

IV. Discussion

From the above analysis we gathered a set of “clear-cut” findings backed up by the evidence reported above. Moreover we observed a few “non-simple to interpret” findings.

A. Evidence-based findings and pragmatic outcomes

We found that modelling is used by a relevant proportion of the respondents (105 corresponding to 68%) and roughly half of them also use some MD* techniques (50, i.e. 32% of respondents).

A surprise comes when we consider that those who assert to have some MD* experience are just 37 (24% of respondents). That could mean that while some practitioners use techniques considered by us as related to MD*, they do not think of themselves as applying comprehensive MD* methodologies, i.e. that difference is due to a partial application of MD*. Another possible explanation for that difference is that a relevant portion of the practitioners is starting right now to experiment MD* techniques so they have not matured one full year of experience yet.

Concerning the key MD* techniques – code generation, model execution, and model transformation – roughly half of the modellers apply code generation, while 15% and 10% of the 105 modellers apply execution and transformations respectively. In particular, we observed a certain dissatisfaction about MD* tools: respondents believe that available tools don’t fit today’s industry needs. Concerning the experience of personnel in MD* techniques: a large number of respondents confessed nearly no experience in MD*, just a few of them had more than 10 years of experience. When we analysed the maturity of companies as a function of their size, we discovered that large and micro companies are more likely to use MD* techniques than small and medium ones.

The evident relevance of software modelling and the low level of maturity in MD* deserves attention from industries and universities. The former should invest more in research (MD* tools and techniques are needed) and training (experts in

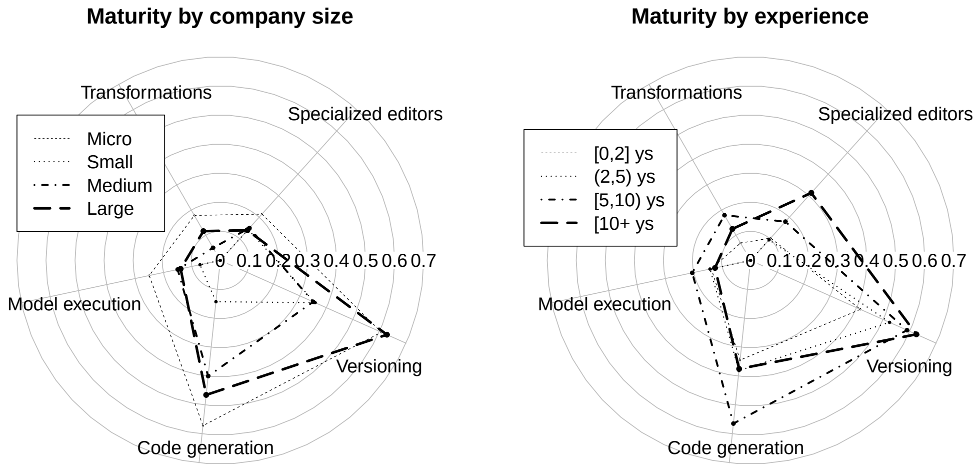

Fig. 5. Maturity with respect to company size (left) and experience in modelling (right)

MD* are needed), and the latter should produce more experts in model driven techniques. This strongly suggests improving university curricula with specific courses dealing with topics related to software modelling, and more specifically with code generation, model execution, and model transformation. Most of the times, students are trained to build new systems using traditional processes and only in the better cases the foundation of MD* are explained in software engineering courses (this is the case in the university of Genova). While it is our opinion that they should focus more on modelling and model driven techniques (in particular in automatic code generation, given that, it is the most used in the industry).

On the university side, the dissatisfaction about tools should be a prompt to produce new prototypes and experiment more in this direction. On the industrial side, we can infer a huge market opportunity for MD* tools; moreover, investments in this market could obtain large returns especially for large companies.

The revealed difference between micro-large and small-medium companies suggests that the latter should prepare for newer technologies (sometimes already used by the former) that will probably help them to become more mature.

B. *Difficult-to-interpret findings*

Together with the evidence-based findings discussed in section IV A, we obtained a set of results difficult to interpret or that deserve a more deep investigation. We list them in the following:

1) A large number of respondents confessed nearly no experience in MD*. See Figure 4 (middle)
2) Code generation is often applied but for a limited extent (i.e., the percentage of generated code is small)
3) Micro companies seem more mature in MD* than larger companies. See Figure 5 (left)

V. Debriefing session

After the data analysis and interpretation of the results, we conducted a debriefing session with three expert software professionals, which participated in our survey, in order to understand MD* findings that are difficult to interpret.

The experts we asked for clarification cover different features of MD*: (1) is the responsible architect for the design of an in-house MD* solution (in short, a suite for the rapid development of informative systems) for a large organization; (2) is the CEO of a company producing a MD* tool for the development of Web applications based on code generation; (3) is the Sales & Marketing Director of a company producing a model driven Web application framework based on run-time execution of models.

Moreover, we took advantage of the availability of such qualified professionals and asked them what they believe is needed to improve the maturity of MD* in the Italian industry.

The outcome of the interviews with the three experts is summarized by the mind map shown in Figure 6. The experts are identified by a number in the mind map: the legend located inside the figure explains which expert corresponds to a given number (e.g., the number two identifies the CEO of the company producing a MD* tool for the development of Web applications). In that Figure, the four main issues are reported by means of a slogan (e.g., *Higher maturity of micro-companies*). Then, for each issue we listed the explanations given by the experts as nodes. The explanations can be supported by one or more experts: e.g., the first explanation of the first issue (i.e, *Small companies can afford simpler, and possibly non very mature, frameworks, ...*) is supported by two different experts while the second (i.e, *More flexible in adopting new processes and technologies*) is supported by just one of the experts (the number two). Only one explanation was supported by all the experts while the others are supported by one or two experts. That should be considered while interpreting the information provided.

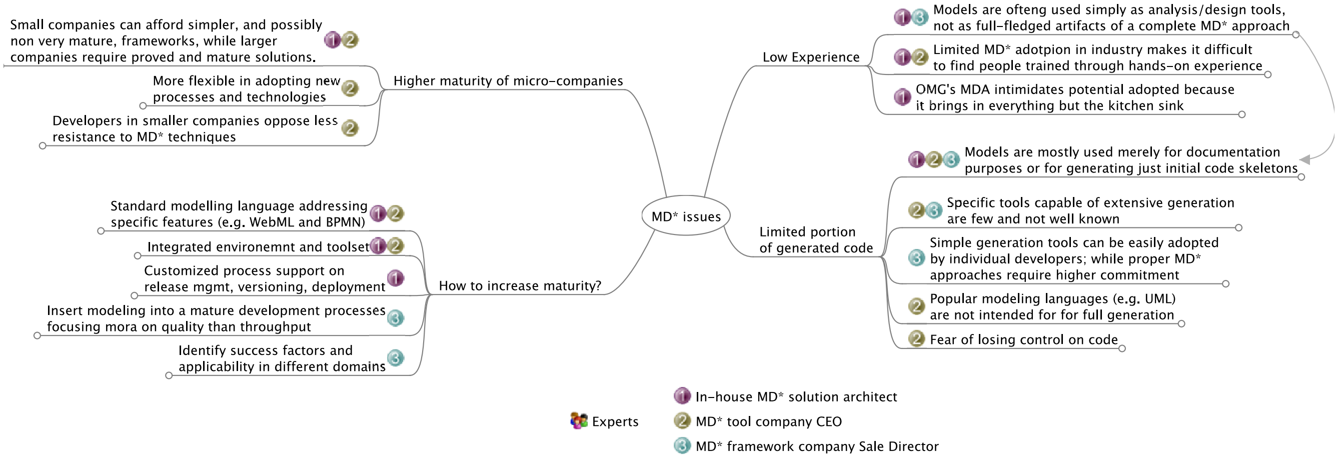

Fig. 6. Mind map of the experts' opinions

A. *Issue 1) Experience in MD* is very low*

Low experience in MD* has been attributed to several different causes: mostly the fact that apparently models are used as tools for documentation or analysis and not as artefacts inserted into a MD* approach. According to our experts, the primary cause for this is little popularity in the industry that limits the capability for developers to build "on-the-job experience". One of the professionals also suggested that huge standardization effort, that bring countless notations and techniques together, may result intimidating and actually preventing diffusion of MD* practices. The considerations from the experts strengthened our consideration about the opportunities in the industry and academia.

B. *Issue 2) The percentage of code that is generated is often low*

Here, returns the supposed use of models barely for documentation purposes or for generating just the initial code skeletons; the typical case consists of generating the code structure from a UML class diagram. In our experts' opinion, the primary cause for this limited use of models lies in the scarcity of appropriate tools and the limited knowledge of the few available. In practice, limited tools are used at individual developer's level because of several factors: it is difficult to get management commitment at team or organization level, common modelling languages (e.g., UML) allow just a limited code generation, and aim for more extensive generation clashes with the fear of losing control over code.

C. *Issue 3) Micro-companies appear to be more mature in MD* than larger companies*

Only two of our experts offered an explanation. Small companies can afford to adopt non fully mature solutions, which are not easily accepted in larger companies, and the "small" size allows more flexibility in using new technologies and processes. Moreover, in large companies there is more resistance, by developers, to the introduction of novel techniques and processes than in smaller ones: novelties threaten personal competence niches which are more likely to be found in large companies.

D. *Question) What is needed to improve the maturity and foster the diffusion of MD* in Italy?*

The experts mentioned factors in three categories: languages, tools, and processes. Standardized languages are the key to the diffusion of MD* approaches; UML and BPMN are positive examples but are not sufficient because they do not cover all the relevant aspects (e.g. interactions and systems communication). Moreover, MD* usage requires integrated toolsets supporting the full development process. From a process perspective, there is a need for customized processes that include not only the generation but also, release management, versioning, and deployment. As far as management is concerned, a successful application of MD* techniques requires understanding the key success factors, the applicability in different domains and the skills required from developers. From a more general perspective, focus on quantitative aspects of software production does not incentive use of models, which can be exploited when quality is considered. In addition, at a management level, it is important to know which are the success factors for different domains and the skills required to practice MD* techniques.

VI. Threats to Validity

We analyse the potential threats to the validity of our study according to the four categories suggested in [21].

Construct validity: the questions we asked are very simple and straightforward (see Table I) therefore we are confident that they actually measured what they were intended for.

Internal validity: the main issues affecting the internal validity of our study concern the framing and sampling of the participants.

- We incurred in a possible selection bias due to the self-exclusion of participants not interested in modelling. Self-exclusion is a well-known problem especially in Internet surveys advertised by means of mailing lists and groups. The possible threat consists in an over estimation of the proportion of respondents who declared interest in modelling and therefore of the overall relevance of modelling and MD* in the Italian industry. However, we believe that the aspects of maturity that we analysed (i.e., level of automation, tools, and experience) are not affected by this threat.

- Another threat derives from the possible "foreign units" in the sample: the target population of our study consisted of development teams, it is possible that the questions were answered by a responded without the required knowledge. We addressed this concern in the protocol: we explicitly required the questionnaire to be filled in by technical personnel involved in the development. Even in case of a knowledgeable respondent, he/she could be unaware of some details; clearly this is more likely if the team is very large [1].
- Eventually, the sampling procedure made it possible to select duplicate units: two different members of the same development team could have answered our questionnaire. We addressed this threat by means of a post-survey validation: we found that the respondents from the same company actually worked in distinct business units and belonged to distinct teams. For this reason, we decided to consider them in the valid sample.

Conclusion validity: most of our analysis was based on simple descriptive statistics, in the few cases where hypothesis testing was opted for non-parametric tests (i.e, Kruskal-Wallis and Mann-Whitney) that can be used without specific assumptions (e.g., without checking data normality).

External validity: we used a non-probabilistic sampling schema since we expected a low diffusion of MD* techniques in the industry and consequently we supposed it was difficult to contact a reasonable number of adopters. This should be considered interpreting the results we obtained: even if the demographics of our sample is quite diverse, the generalization of our results to the entire population may not be appropriate. Moreover, given the sampling strategy we adopted we can not calculate the response rate (this is also common in other software engineering surveys [1]).

VII. Related work

In literature, empirical studies — i.e., industrial surveys, controlled experiments, case studies and systematic reviews — evaluating MD* and considering aspects such as: benefits of its adoption, maturity and real usage in the industry are really rare [14].

Mohagheghi et al. [14] present a literature review of empirical studies (and more in general of experiences) about MDE in industry. The period considered is 2000-2007 and the goal is evaluating MDE benefits and possible limitations. They studied the maturity in the adoption of MDE using level of automation and tools as dimensions; we took inspiration from them in this paper. To evaluate the level of automation, they take into account (as we did), automatic generation of code and executable models. They conclude that: the current state of MDE cannot be considered mature (this is in line with the current situation in Italy), traditional software processes does not fit well with MDE and they must be adapted to it. Moreover, complexity of models is a challenge itself. Other interesting findings of this review can be summarized as follows: (i) the use of MDE is not restricted to a little number of domains; (ii) there is some evidence of a productivity gain with MDE, but this evidence is not supported by the data.

Forward et al. [4] analyse the results of a survey on the perception of software modelling by software practitioners. It mainly investigates how, when and why software developers use, or not, models and in the case they use it, with which notations and tools. Primary findings are: (i) developers consider models in a broader sense (i.e., not only UML models but also textual DSL models), (ii) UML is the predominant modelling notation (the same is true in [18]) but is often used informally and (iii) modelling tools are mainly used for documentation (in line with our results) and (iv) it is uncommon that models are used for generating code.

Hutchinson et al. [7] present the results of an empirical study on the assessment of MDE in industry, having the goal of understanding the reasons of success or failure of MDE. Three forms of investigation are used: questionnaires, interviews and on site observations. The questionnaire has received over 250 responses. Main results can be summarized as follows: almost two-thirds of the respondents believe that the use of MDE has improved productivity (code generation is the most important aspect of MDE productivity gain) and maintainability. On the other hand, most of the respondents think that the use of MDE implies extra training and tools, the latter needed for a practical deployment of MDE. UML is the most used language (the same is true in [18]) and a good number of respondents use developed in-house DSLs.

Instead, the paper [6] focuses on the deployment of MDE in industry. It illustrates three industrial case studies and identifies some lessons learned; in particular the importance of complex organizational, managerial and social factors in the success or failure of the MDE deployment. Among the successful factors, we can mention: (i) progressive and iterative approaches; (ii) a strong motivation (user and organization must be motivated to use the new strategy); (iii) organizational adaptability/flexibility (the organization must be flexible and responsive to adapt process and way of doing business); and (iv) a clear business focus. Another important factor, reported in this work, is the management of concurrent changes due to MDE deployment.

Heijstek et al. [5] study the impact of MDD in a large scale industrial project. The sources of information are data from the project repository and semi-structured interviews with team members. Conclusions are that: almost two-thirds of the total effort is spent on developing models and that the team members relate an increase in productivity, besides a perception of improvement of the overall quality and a reduction of complexity. The authors confirm the increase of quality by counting the average number of defects in relation of the average number of defects found in similar size projects in which MDD was not used.

Authors in [16] present a model that can be used for identifying the level of MDD maturity of an organization. Five maturity levels are identified, each of which is characterized by a coherent set of engineering, management and support practices over a set of MDD artefacts called MDD elements. Maturity levels starts from level 1 (ad-hoc modelling), the less mature MDD adoption level, and they ends at level 5 (ultimate MDD), the more mature MDD adoption level. In our work we used substantially the same MDD elements (models, model transformations, code generations and tools) of [16] but without the explicit definition of MDD practices nor maturity levels. Although we agree with the proposed five level taxonomy, in our work we did not adopt it because of the objective classification difficulty.

VIII. Conclusions and future work

In this paper, we presented some results from a survey performed to investigate what is the level of maturity of MD* in the Italian industry.

While we found that the practice of producing models is quite widespread (68% of the entire sample), the percentage of development teams using MD* techniques is significantly smaller (48% of the adopters of modelling). Apparently, many companies use models to capture an high level view of the system and for documentation purposes. A possible cause of this reality is that developers are not enough educated in model-driven techniques. Moreover, several problems emerged concerning the tools supporting MD*. Experts agree that a set of standard languages, easy to learn and to use, would improve the diffusion of MD*. Probably the ideas behind MD* should be reorganized and repackaged into a more immediate “turnkey solution”. The whole MDA effort[9], while complete and consistent, looks like a “bulky monolith” and may result intimidating to novices.

As future work we would like to compare the level of maturity of MD* in Italian companies to the situation in other countries replicating this study in other nations.

Acknowledgements

We would like to thank the people who dedicated their time to take part in our survey and the organizers of CodeMotion for the help provided to spread our survey. We would like also to thank the experts which we interviewed and discussed our findings with. They are: Paolo Arvati from CSI Piemonte, Stefano Butti from WebRatio and Paolo Predonzani from ManyDesigns.

References

[1] S.L. Pfleeger B. Kitchenham. Personal opinion surveys. In Forrest Shull and Singer, editors, *Guide to Advanced Empirical Software Engineering*, pages 63–92. Springer London, 2008.

[2] V. Basili, G. Caldiera, and D. H. Rombach. *The Goal Question Metric Paradigm, Encyclopedia of Software Engineering*. John Wiley and Sons, 1994.

[3] M. Ciolkowski, O. Laitenberger, and S. Biffl. Software reviews: The state of the practice. *IEEE Software*, 20(6):46–51, November/December 2003.

[4] A. Forward, O. Badreddin, and T. C. Lethbridge. Perceptions of software modeling: A survey of software practitioners. In *5th Workshop from Code Centric to Model Centric: Evaluating the Effectiveness of MDD (C2M:EEMDD)*, 2010.

[5] W. Heijstek and Michel R. V. Chaudron. Empirical investigations of model size, complexity and effort in a large scale, distributed model driven development process. In *Proceedings of the 2009 35th Euromicro Conference on Software Engineering and Advanced Applications*, SEAA ’09, pages 113–120, Washington, DC, USA, 2009. IEEE Computer Society.

[6] J. Hutchinson, M. Rouncefield, and J. Whittle. Model-driven engineering practices in industry. In *Proceedings of the 33rd International Conference on Software Engineering*, ICSE ’11, pages 633–642, New York, NY, USA, 2011. ACM.

[7] J. Hutchinson, J. Whittle, M. Rouncefield, and S. Kristoffersen. Empirical assessment of MDE in industry. In *Proc. of the 33rd International Conference on Software engineering*, ICSE ’11, pages 471–480, New York, NY, USA, 2011. ACM.

[8] D. Damian J. Singer, M.A. Storey. Selecting empirical methods for software engineering research. In Forrest Shull and Singer, editors, *Guide to Advanced Empirical Software Engineering*, pages 285–311. Springer, 2008.

[9] A. Jelitshka, M. Ciolkowski, C. Denger, B. Freimut, and A. Schlichting. Relevant information sources for successful technology transfer: a survey using inspections as an example. In *First International Symposium on Empirical Software Engineering and Measurement, 2007. (ESEM 2007)*, pages 31–40. IEEE, September 2007.

[10] T Kapteijns, S Jansen, S Brinkkemper, H Houet, and R Barendse. A comparative case study of model driven development vs traditional development: The tortoise or the hare. *From code centric to model centric software engineering Practices Implications and ROI*, page 22, 2009.

[11] A. G. Kleppe, J. Warmer, and et al. *MDA Explained: The Model Driven Architecture: Practice and Promise*. Addison-Wesley Longman Publishing Co., Inc, 2003.

[12] T. C. Lethbridge. A survey of the relevance of computer science and software engineering education. In *Proceedings of the 11th Conference on Software Engineering Education and Training*, pages 0056–, Washington, DC, USA, 1998. IEEE Computer Society.

[13] S. Mellor and M. Balcer. *Executable UML: A foundation for model-driven architecture*. Addison-Wesley, 2002.

[14] P. Mohagheghi and V. Dehlen. Where is the proof? - a review of experiences from applying MDE in industry. In *Proc. of the 4th European conference on Model Driven Architecture: Foundations and Applications*, ECMDA-FA ’08, pages 432–443, Berlin, Heidelberg, 2008. Springer-Verlag.

[15] P. Mohagheghi, M.A. Fernandez, J.A. Martell, M. Fritzsche, and W. Gilani. MDE adoption in industry : Challenges and success criteria. *Models in Software Engineering*, pages 54–59, 2009.

[16] E. Rios, T. Bozheva, A. Bediaga, and N. Guilloreau. MDD maturity model: A roadmap for introducing model-driven development. In Arend Rensink and Jos Warmer, editors, *Model Driven Architecture Foundations and Applications*, volume 4066 of *Lecture Notes in Computer Science*, pages 78–89. Springer Berlin / Heidelberg, 2006.

[17] M. Torchiano, M. Di Penta, F. Ricca, A. De Lucia, and F. Lanubile. Migration of information systems in the italian industry: A state of the practice survey. *Inf. Softw. Technol.*, 53:71–86, January 2011.

[18] M. Torchiano, F. Tomassetti, A. Tiso, F. Ricca, and G. Reggio. Preliminary findings from a survey on the MD* state of the practice. In *International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 372–375, 2011.

[19] Markus Völter. MD* best practices. *Journal of Object Technology*, 8(6):79–102, 2009.

[20] D. S. Walonick. *Survival Statistics*. StatPac, Inc., 1997.

[21] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering - An Introduction*. Kluwer Academic Publishers, 2000.

[9] In this case, we intend exactly MDA, i.e., the registered trademark of OMG recommending the usage of OMG standards (e.g., UML)